The Tiny Tim User's Guide

Version 6.0
July 2001

http://www.stsci.edu/software/tinytim

John Krist (krist@stsci.edu)
Richard Hook (rhook@eso.org)

Changes in Tiny Tim Version 6.0

General

Tiny Tim now writes out only FITS files, and these files have ".fits" extensions. Log files are no longer produced, as the FITS header now contains the PSF information.

A new interpolation method (damped sinc) is now used, which is faster than the old way (adaptive sinc). This can increase Tiny Tim's speed by up to a factor of 7 for large PSFs.

User-defined spectrum files must now begin with a keyword indicating the units of the flux column (PHOTLAM, FNU, FLAM, JY). Any fluxes in these units will be converted into the internal form used by Tiny Tim.

Power laws are now available for specifying the object spectrum.

STIS

The default focus offset for the imaging modes has been updated based on phase retrieval analysis.

ACS (Advanced Camera for Surveys)

Many significant changes have been made to support the ACS, including : field-dependent aberrations derived from ray-trace analysis, HRC CCD red halo, HRC off-spot coronagraphic PSFs, geometric distortion, user-defined scene PSF convolution and distortion. A new program, *tiny3*, has been added specifically to process ACS PSFs and simulated scenes. This new program is documented in the ACS section of this manual.

<u>Contents</u>

Introduction

Tiny Tim is a software package for generating Hubble Space Telescope (HST) model point spread functions (PSFs). It is easy to use and relatively fast. It is written in standard C, and thus is portable to a variety of computers and operating systems, including UNIX, Linux, and Windows.

For details on how PSFs are generated, please see Appendix B.

For the most up-to-date information on Tiny Tim, as well as example applications of PSF models and the FAQ list, check out **http://www.stsci.edu/software/tinytim**.

Tiny Tim is a personal endeavour of John Krist, with additions by Richard Hook (initial NICMOS support and user-defined filter tables). The Space Telescope Science Institute, which is the distribution point for Tiny Tim, *is not* responsible for the software. Any questions or comments, however, may be sent to krist@stsci.edu.

Uses of Model PSFs

Besides optical analysis and deconvolution, model PSFs have been used for a wide variety of applications:

- **Proposal planning :** determining peak pixel values relative to total PSF flux; predicting the width of the PSF core at a given wavelength
- **Photometry :** PSF profile fitting by the shifting and scaling of models; testing completeness by generating synthetic images
- **Algorithm testing :** photometric and deconvolution procedure testing
- **PSF subtraction**
- **Data modeling :** convolution of model data with model PSFs to simulate observed images

Caveats

Model PSFs do have significant limitations, though some of these are shared with observed PSFs as well. The simulation is only as good as the input parameters, which are often inaccurate due to our limited knowledge of the actual, time-varying optical system. A large number of factors affect the PSF, including:

- Filter, optical, and detector responses with wavelength
- Object spectrum
- Focus (time and position dependent)
- Aberrations (which are generally time, focus, and position dependent)
- Polishing errors in the mirrors
- Obscuration positions (field dependent) and sizes (focus dependent)
- Detector orientations (rotation, pixel sizes)
- Detector subpixel responses
- Geometric distortions
- Scattering by dust, scratches, chips, rough surfaces
- Scattering by the detector

Many of these vary with time and/or field position, so that it is impossible to create a model that will match an observed PSF pixel for pixel. Of course, even observed PSFs will often have significant differences due to these same variations.

Installation

Compiling Tiny Tim

Tiny Tim is available for download from the World Wide Web or via FTP. To get it from the Web, go to **http://www.stsci.edu/software/tinytim**. Using FTP, go to `ftp.stsci.edu` and login anonymously, then cd to software/tinytim.

Tiny Tim is distributed in a compressed tar file as source code, with supporting ASCII tables and FITS files. This file should be downloaded in binary mode and then unpacked into a separate directory on your machine, like so:

```
uncompress tinytimv60.tar.Z
tar xvf tinytimv60.tar
```

After this, the program needs to be compiled. A `Makefile` is provided to automate this task, with default compiler settings for Solaris, Linux, and HP versions of UNIX. For other systems, you should edit the `Makefile` and change the compiler settings for the OTHER mode as necessary. The programs are compiled by running **make** using one of the following:

```
make sparcstation
make linux
make hp
make other
```

Setting Pointers to Tiny Tim

The appropriate pointers to the Tiny Tim programs and directory must be included in your environment file (`.cshrc`, `.profile`, or whatever is appropriate to the system/shell).

The following shows a c-shell example (appropriate for users at STScI):

```
setenv TINYTIM /usr/stsci/tinytim
alias tiny1 $TINYTIM/tiny1
alias tiny2 $TINYTIM/tiny2
alias tiny3 $TINYTIM/tiny3
```

If you are running the Bash shell (the default for most Linux installations), you can follow this example for entries in your `.bash_profile` file (set the `TINYTIM` directory to whatever is appropriate for you) :

```
TINYTIM=/usr/local/tinytim
export TINYTIM
alias tiny1='$TINYTIM/tiny1'
alias tiny2='$TINYTIM/tiny2'
alias tiny3='$TINYTIM/tiny3'
```

Multithreading

As of version 5.0, Tiny Tim supports multithreaded execution on multiprocessor systems that support POSIX threads (Pthreads). This includes Solaris 2.5 and later, and certain distributions of Linux (Red Hat 5.2 or later, otherwise check **http://www.uk.linux.org/SMP/title.html** for more info).

To compile Tiny Tim for multiprocessor systems, use one of the following `make` commands :

```
make threadedlinux
```
or
```
make threadedsolaris
```

Note to Linux users : *If you have a multiprocessor Linux system, the kernel must be recompiled to support SMP (symmetric multiprocessing) to use more than one processor. The default installation for Red Hat, for instance, has SMP disabled. If you ordered a multiprocessor system with Linux installed, it is probably already SMP enabled. It is recommended that you run kernel 2.0.36 or later.*

Running Tiny Tim

Tiny Tim consists of three programs – *tiny1*, *tiny2*, and *tiny3*. *tiny1* asks a series of basic questions and generates a parameter file that is read by *tiny2*. *tiny2* does the actual PSF modeling, requiring no additional input from the user. *tiny3* geometrically distorts ACS PSFs and input scenes (it is only used for ACS modeling). *tiny2* and *tiny3* can be run in batch mode if desired. All three programs require at minimum a single command line argument – the name of the parameter file. For example:

```
tiny1 test.par
tiny2 test.par
tiny3 test.par
```

tiny2 produces a FITS image file containing the PSF. If you are modeling the ACS, *tiny3* will create a FITS file containing the geometrically distorted PSF or PSF-convolved scene (see the ACS section for more details).

Optional Command Line Parameters

tiny1 allows the following optional command line parameters (given after the parameter filename). *tiny3* also has additional command line parameters, and those are listed in the ACS section.

ebmv=*x* *or* **Av=***x*
> Interstellar extinction of *x* magnitudes is applied to the object spectrum, specified in the form of E(B-V) or visual extinction (A_v). Do not specify both.

filter=*filename*
> The user can provide his own system throughput curve by providing an ASCII file which contains wavelength (in nm) and throughput pairs, one per line. This will be used instead of the default filter curves. See Appendix C : *Object Spectra and Filter Curves*.

jitter=*x*
> Convolve the PSF with a Gaussian kernel approximating *x* milliarcseconds RMS of jitter.

spectrum=*filename*
> The user can provide his own spectrum by providing an ASCII file which contains wavelength (in Angstroms) and flux pairs, one per line. See Appendix C : *Object Spectra and Filter Curves*.

wmag=*x*
> Increase or decrease by a factor of *x* the default number of wavelengths used in computing a polychromatic PSF (ie. `wmag=0.5` would use only half the default number of wavelengths, while wmag=2 would double the number, more finely sampling the response curve). Increasing wmag provides a somewhat smoother PSF at radii beyond about 3". A minimum of one wavelength will be used.

For example,

```
tiny1 test.par ebmv=1.4 jitter=15.0
```

General Input Parameters

A number of the questions which *tiny1* asks are common to all of the instruments and are discussed here. Inputs specific to each camera are listed in later sections.

PSF Size

tiny1 asks you for the diameter of the PSF you wish to generate in arcseconds. From this it determines the proper grid sizes to use. The maximum possible size of a model PSF is limited by the array dimensions used to compute the Nyquist-sampled PSF. At short wavelengths, it may not be possible to compute a PSF larger than 7". Generally, the models are not very good past a radius of ~2", due to the effects of scatter and high-frequency aberrations.

Generating a large PSF when you really only need a smaller PSF will simply waste computer cycles. Note that *tiny1* suggests a minimum good PSF size. If the camera has PSFs with halos (ACS/HRC, or F1042M filter halo in WFPC2), then you should use at least this minimum diameter.

Filters/Wavelengths/Spectrum

You have the option of generating a purely monochromatic PSF or one that includes wavelength variations over a filter's bandpass. Monochromatic PSFs are suitable for modeling narrow-band filters or the linear ramp filters in WFPC2 or ACS.

Tiny Tim has a database of wavelengths and weights for each filter. A polychromatic PSF is created by generating monochromatic models at each wavelength and adding them together with the corresponding weights. The wavelengths have been chosen to properly sample the filter's bandpass. The throughput curve is multiplied by a spectrum to simulate the PSF variations caused by object color. You can apply interstellar extinction using the *ebmv=x* or *Av=x* keyword parameter on the *tiny1* command line. See Appendix C : *Object Spectra and Filter Curves* for more information.

You also have the option of providing a list of wavelengths and weights instead of using the default filter curves and spectra. See Appendix C : *Object Spectra and Filter Curves* for more information.

Sampling

Model PSFs can be generated at an arbitrary sampling. *tiny1* will ask if you would like to generate a subsampled PSF (except for the ACS; see the ACS section for details). If you choose to do so, it will ask you for the subsampling factor, which is an integer from 1 to 10 (*e.g.* 5 means that the resulting pixels are 1/5 of normal size in each dimension). Subsampling generally increases the computation time by the square of the subsampling factor for a given PSF diameter. If you are using high subsampling factors (>5), then you may be better off subsampling by 5, for instance, and then interpolating the result.

Subsampling is useful in deconvolution, photometry, and PSF subtraction. By subsampling the PSF, interpolating, and then rebinning it, the model can be registered to high precision – especially important for highly undersampled cameras like the WFCs in WF/PC-1 and WFPC2. When subsampling is used for WFPC2, STIS CCD, or the ACS HRC/WFC, you need to manually convolve the PSF with a kernel, *after rebinning to normal sampling*, that simulates blurring caused by charge diffusion (described in the sections on these respective cameras).

Note that the PSF is centered in the middle of a pixel, whether that pixel is normally sampled or subsampled. The center pixel is *n*/2, where *n* is the number of pixels along one edge of the final image (beginning at 0,0).

Jitter

The pointing stability of the telescope in fine lock is typically better than 5 mas RMS. In the first year or so, the pointing was worse, ~ 15 mas. Typically, any value under 7 mas is not noticeable. For most cases, you should not add any. If you want jitter, you can specify the amount on the *tiny1* command line, like so :

```
tiny1 test.in jitter=7
```

where the amount of jitter is given in milliarcseconds RMS (Gaussian).

Output Files

The model PSF will be written to a FITS image file by *tiny2*. If you generated just one PSF, the file name will have the rootname you specified in *tiny1* with "00" appended (*e.g.* `nicmospsf00.fits`). If you generated PSFs for more than one position (an option available for WF/PC-1, WFPC2, NICMOS, and ACS), the other files will have successive numbers (`nicmospsf01.fits`, `nicmospsf02.fits`, …).

Note that when modeling the ACS, the PSFs produced by *tiny2* have a "_psf.fits" extension (*e.g.* `acs00_psf.fits`). These PSFs are intended for use by *tiny3* and not the user. Then, the PSFs produced by *tiny3* will have "normal" Tiny Tim filenames (*e.g.* `acs00.fits`).

An Example Run

As an example, let's generate a WF/PC-1 WFC2 PSF in F555W at different positions on the detector. User input is in ***bold italics***.

```
% tiny1 wfpc1.in

        ------ Tiny Tim v6.0 : The HST PSF Generator ------

            Developed by John Krist (krist@stsci.edu)
        Additional support by Richard Hook (rhook@eso.org)
         ** http://www.stsci.edu/software/tinytim.html **

--------- Aberrated Cameras ---------        -------- Corrected HST Optics -------
1) Wide Field Camera (WFPC1) (f/12.9)     5) WFPC2 - Wide Field Camera (f/12.9)
2) Planetary Camera  (WFPC1) (f/30.0)     6) WFPC2 - Planetary Camera  (f/28.3)
3) FOC f/48                               7) COSTAR corrected FOC f/48 (f/72.5)
4) FOC f/96                               8) COSTAR corrected FOC f/96 (f/151)

---- Second Generation ----                  -------- Third Generation ---------
 9) NICMOS Camera 1 (f/80)                15) ACS - Wide Field Channel (f/26)
10) NICMOS Camera 2 (f/45)                16) ACS - High Resolution Channel (f/72)
11) NICMOS Camera 3 (f/17.2)              17) ACS – HRC Coronagraph off-spot PSF
12) STIS CCD (f/36)                       18) ACS - Solar Blind Channel (f/72)
13) STIS NUV-MAMA (f/90)
14) STIS FUV-MAMA (f/90)

Choice : 1

Enter Chip (1,2,3,4) : 2
```

Next we'll be asked for the positions of the PSFs on the detector, which must be integer numbers. This question is only asked for WF/PC-1, WFPC2, NICMOS, and ACS. Let's assume that we want to generate PSFs at three different locations on WFC2. We'll create a plain text file `xy.lis` which lists the coordinates, one pair per line, like so (up to 100 PSFs are allowed) :

```
100 100
400 400
700 700
```

We'll enter the filename, preceeded by a "@" for the position.

```
Enter position (x & y) on detector in INTEGER
pixels (range = 0-799) or the filename of a list
of positions preceded by a '@' (ie. @xy.lis).

Position on detector : @xy.lis
```

Now *tiny1* is going to ask for the date of observation. It only asks for the date for WF/PC-1, the pre-COSTAR FOC, and NICMOS.

```
Enter the date of observation in the form dd mm yy.
For example, 6 April 1991 would be "6 4 1991".
Date : 4 4 1992

Date is April 4, 1992
```

We are now asked for the filter. We could say "**mono**" if we wanted to generate a purely monochromatic PSF. If we had provided our own filter throughput table (using the *filter=x* option on the command line), we would not be asked this question. In this example, we'll just use the filter information which is in the Tiny Tim database.

```
Select filter passband :
    - Enter the name of the filter (eg. f555w)
    - Enter MONO to specify a single wavelength
Choice : f555w
```

If we chose something other than a narrowband filter (which has only one wavelength), and did not choose the MONO option, then we must select an object spectrum. You can choose from a list of 17 spectra ranging from O to M type stars, use a blackbody or power-law spectrum, or provide your own spectrum (see Appendix C: *Object Spectra and Filter Curves*). In this example, we'll choose one of the provided spectra.

```
Choose form of object spectrum :
    1) Select a spectrum from list
    2) Blackbody
    3) Power law : F(nu) = nu^i
    4) Power law : F(lambda) = lambda^i
    5) Read user-provided spectrum from ASCII table
Choice : 1
```

A list of the available spectra, along with their color terms, are displayed and we choose one of them.

Now we have to specify how large a PSF to generate. The maximum size depends on the wavelength and focal ratio of the camera. In this case, we want to generate a 6 arcsecond diameter PSF (actually, the closest we can get to 6", given the finite size of the detector pixels).

```
Assuming detector pixel size of 0.09993 arcsec and f/ratio of  12.9

The maximum computable PSF size is 15.00 arcsec.
The recommended size is about 6.0 arcseconds.
What diameter should your PSF be (in arcseconds)? : 6.0
```

We want normal sampling in this case.

```
Do you wish to generate a subsampled PSF? (Y/N) : n
```

We want the rootname of the output FITS files to be `wfpc1` :

```
Rootname of PSF image files (no extension) : wfpc1

To execute the next stage, enter the command :
        tiny2 wfpc1.in
```

tiny1 will write out a file called wfpc1.in. We'll feed it to *tiny2*:

    tiny2 wfpc1.in

tiny2 will compute a PSF for each position in our list and write it out to a FITS file. In this case, the filenames will be `wfpc100.fits,` `wfpc101.fits,` and `wfpc102.fits.`

General PSF Properties

Focus Changes (Breathing and Desorption)

The focus of the telescope varies within an orbit due to thermal effects, which change the position of the HST secondary mirror by 1 – 10 μm on average. This is called *breathing*.

A long-term focus change is caused by desorption of the telescope's graphite epoxy truss. After being placed in orbit, the structure began outgassing, causing it to shrink and the focus to vary. Initially, the rate of desorption was great. In order to remain at best focus, the secondary mirror had to be moved away from the primary in order to compensate for the shrinkage. These mirror moves occurred on a nearly monthly basis during the first two years of operation. During these intervals, however, the focus continued to change in a predictable way (exponential decay). By the time of the first servicing mission, the rate of desorption was low, and mirror moves became less frequent. Today, a mirror move occurs roughly every six months. Between these times, the changes in focus due to breathing outweigh those caused by desorption.

In the aberrated cameras (WF/PC-1 and FOC without COSTAR), breathing causes the positions of the tendrils, rings, and central obscuration shadows in the PSFs to vary, which could lead to significant residuals in deconvolutions. For these instruments, Tiny Tim has a database of dates of mirror moves and the measured focus values for those times. Using a date provided by the user, the information in this database, and the power law describing desorption, the proper focus for that time is computed (excluding breathing).

For the corrected cameras except NICMOS (WFPC2, FOC+COSTAR, STIS, ACS), time-dependent focus is *not* included, so you are not asked for a date. The PC in WFPC2 is assumed to be in perfect focus at the center of the field, as is FOC/COSTAR and the ACS. In NICMOS, focus is computed based on the PAM (Pupil Alignment Mechanism) positions of the observation and of best focus for the camera.

In all cases, adjustments to focus (to simulate breathing, for instance) can be made by editing the file produced by *tiny1*. Towards the end of that file is a list of aberrations, one of which is focus. These values are in RMS waves of the aberration at 547 nm. To simulate breathing, 0.011 should be added or subtracted to the current value for each micron of secondary mirror despace (subtracting means that the secondary moved towards the primary).

Wavelength

The largest variations in the PSFs occur with wavelength. As wavelength increases, the diffraction structures (rings, spiders) expand proportionally. In wide-band filters, the expansion of the PSF over the bandpass will blend diffraction rings, creating smooth wings. The PSF may even be slightly different due to color differences among objects in the same filter. In narrow-band filters, though, the diffraction rings are sharp, and the bandpass is too narrow to show color effects.

As wavelength decreases, the PSF becomes more sensitive to aberrations, including zonal errors in the mirrors. Thus, a PSF in the UV will look completely different from one in the near-IR.

Camera Specifics

WFPC2

The second Wide Field and Planetary Camera (WFPC2) is a replacement for WF/PC-1 that corrects for the spherical aberration in the Hubble Space Telescope. WF/PC-1 had a rotating pyramid mirror that selected either four wide field or four planetary channels. The pyramid in WFPC2 is fixed, so only four channels are available – three wide-field cameras (WFC) and one planetary camera (PC). Each channel has a Cassegrain repeater system that has a three-vane spider. The CCDs are 800 x 800 element devices with 15^2 μm pixels. WFPC2 operates in the wavelength range of 150 – 1100 nm.

WFPC2 Optical Parameters

Camera	Focal Ratio	Pixel Size Arcsec	Field Size Arcsec
PC1	28.3	0.0456	35
WF2,3,4	12.9	0.1	76

Tiny1 Queries Specific to WFPC2

- **Camera** : either 1 (for the PC) or 2-4 for the WFCs
- **Position of PSF on the detector** : used to compute the proper obscuration function and determine field dependent aberrations; the name of a file containing up to 100 coordinate pairs can be entered, preceded by a "@", and *tiny2* will generate a PSF for each position, with output filenames like test00, test01, …

Field Dependence and Aberrations

Like WF/PC-1, the obscurations in the repeaters (spiders & secondaries) are not in a plane conjugate to the entrance pupil. This causes the pupil pattern to vary with field position, with the WFPC2 obscurations appearing to shift relative to the telescope's. This creates significant variations in the PSF depending on where the object is imaged on the detector, including changes in the shapes of the diffraction rings, "squishing" of the core, and alteration of the banding patterns in the diffraction spikes. The rotation of the detector and obscurations are different for each camera, so the PSFs are not interchangeable.

In addition to obscuration shifting, some aberrations vary with field position. These include focus, coma, and astigmatism. Jon Holtzman (NMSU) has provided the field dependence relations for these aberrations.

Observed PSFs in the F1042M filter have a halo around them unique to that filter (see Krist 1999). An empirical model of the halo is added to model PSFs.

Geometrical distortion has minor effects on the WFPC2 PSF, as the pixel scale changes slightly towards the field edges. It is greater in the WFCs than in the PC.

Each camera has a slightly different focus than the others. Attempts are made to keep the PC in focus, due to its higher resolution. The WFCs are then slightly out of focus, with the worst, WFC3, having a defocus equivalent to about 9 μm of breathing.

Tiny Tim accounts for both obscuration shifting and aberration variations with field position, using the coordinates entered by the user. Geometrical distortion is *not* included. The focus offsets between the cameras are included, with the PC assumed to be at perfect focus.

Large Angle Scattering

Some of the light from the target will be scattered by the CCD surface and reflected back by the optics, distributing flux further out into the wings. Tiny Tim does not include this effect, so the PSF models will underestimate the flux in the wings.

The scattering is not uniform, but rather consists of streaks and rays radiating from the core. These change intensity and position with the location of the object on the detector, further adding to the residuals one can expect when subtracting models from observed PSFs.

Subpixel Response Function

All CCDs suffer from an effect called *charge diffusion*. CCD pixels are not physically separate elements; they are defined by the electromagnetic fields created by the electrode structure on the detector surface. Thus, pixel edges are actually vague. When a photon is converted to an electron in the substrate, the electron tends to be attracted to the nearest electrode. But in regions where the field strength is weak (i.e. furthest from the electrode), an electron may travel some distance from where it was generated, even into an adjacent pixel (hence the term charge diffusion). The net effect is blurring of the image, as some photons hitting within one pixel are detected in another. Since this causes a dependence on where in the pixel the photon hit, it is characterized by the *subpixel response function*.

The amount of charge diffusion varies with wavelength. Shorter wavelength photons are converted to electrons near the illuminated surface of the detector, while it typically requires a longer path for long wavelength photons (very long wavelength light can actually pass through the substrate without being detected). In backside illuminated CCDs (like in WF/PC-1, STIS, and the upcoming Advanced Camera), the electrodes are on the rear side of the detector, closer to where long wavelength photons are converted. Thus, in such devices charge diffusion becomes worse with decreasing wavelength.

For front-side illuminated CCDs, like those used in WFPC2, the opposite is *not* true. Charge diffusion is high at longer wavelengths (>~800nm), but also at shorter ones (<~400 nm), since the short wavelength photons are scattered by the electrode structure on the illuminated surface. Charge diffusion is minimal (but not insignificant) at about 550 nm.

Charge diffusion in WFPC2 slightly blurs the images, producing an effect similar to jitter. In the PC it is equivalent to about 14 mas of jitter, while in the WFCs, because they largely undersampled the PSF, the effect is more like 40 mas. The wavelength dependence of the subpixel response function has not been characterized except in limited laboratory tests.

Tiny Tim approximates the effects of charge diffusion by convolving the normally sampled WFPC2 PSF by a Gaussian kernel of 3 x 3 pixels. This kernel is most appropriate in the range of 500 – 600 nm, but since nothing better is available, it is used at all wavelengths. ***Because this effect occurs at the pixel level, this charge diffusion kernel is <u>not</u> applied to subsampled PSFs. You must apply the kernel <u>after</u> rebinning the subsampled PSF to normal sampling.*** Ideally, a subsampled subpixel response function would be convolved with a subsampled PSF and then sampled to get a PSF at the correct pixel phase, but this process is not practical for most uses of the models.

The kernel used by Tiny Tim for WFPC2 is:

0.0125	0.05	0.0125
0.0500	0.75	0.0500
0.0125	0.05	0.0125

A recent paper (Lauer 1999) describes in further detail the subpixel response of WFPC2.

Filters

WFPC2 has a wide variety of narrow, medium, and wide-band filters. It also has quadrant filters (like methane and redshifted O II) and polarizers. In some modes, the polarizers and the Woods filters (F160AW and F160BW) vignette part of the field, effects that are *not* included in Tiny Tim. There are also linear ramp filters, which are not explicitly included in Tiny Tim but can be modeled by generating a monochromatic PSF of the appropriate wavelength (by selecting MONO when asked for the filter name).

The available WFPC2 filters in Tiny Tim are:

F122M	F130LP	F157W	F160BW	F165LP
F170W	F185W	F218W	F255W	F300W
F336W	F343N	F375N	F380W	F390N
F410M	F437N	F439W	F450W	F467M
F469N	F487N	F502N	F547M	F555W
F569W	F588N	F606W	F622W	F631N
F656N	F658N	F673N	F675W	F702W
F785LP	F791W	F814W	F850LP	F953N
F1042M	FQUVNA	FQUVNB	FQUVNC	FQUVND
FQCH4NA	FQCH4NB	FQCH4NC	FQCH4ND	

F1042M Halo

Tiny Tim v5.0 and later now includes an approximation to a halo seen only in F1042M images (see the *WFPC2 Instrument Handbook*). This halo contains about 20% of the total flux. Its cause is unknown but is likely due to the fact that the CCD essentially becomes transparent at such long wavelengths and light is scattered within the CCD backing. The halo *is not* seen in any other filters.

Caveats

One of the most popular uses of Tiny Tim WFPC2 PSF models is the subtraction of stars and stellar-like galaxy cores in order to see faint material or companions which may be hiding under the PSF wings. While the models can subtract much of the light, significant residuals will be left behind. Changing the focus may help some, but the main problem is that scatter from the CCD surface is not modeled, due to its very complex nature.

The banding patterns in the diffraction spikes do not match between the model and observed PSFs. These bands are very sensitive to obscuration positions and sizes, focus, and wavelength. PSFs should NEVER be normalized based on the spider intensities, whether using observed or model PSFs.

NICMOS

NICMOS (Near-Infrared Camera and Multi-Object Spectrometer) is optimized for imaging in the wavelength range of 0.9 – 2.5 μm. It consists of three separate cameras, whose characteristics are listed in the table below. Each has a 256 x 256 detector with 40^2 μm pixels.

NICMOS Optical Parameters

Camera	Focal Ratio	Pixel Size arcsec	Field Size arcsec
1	80	0.043	11.0
2	45	0.076	19.55
3	17.2	0.2	51.2

Each pixel is essentially a separate detector – there is no cross-talk between elements. So, unlike WFPC2, there is no charge diffusion that would cause blurring at the pixel level. There is a subpixel variation in response, but it does not appear to significantly affect the imaging performance of cameras 1 and 2. There have been reports of large Camera 3 photometric variations with position within a pixel (see Lauer 1999 and Storrs et al. 1999).

Camera 2 has a coronagraphic mode that is *not* modeled by Tiny Tim.

Tiny1 Queries Specific to NICMOS

- **Location of the PSF on the detector** : used to compute field dependent aberrations; the name of a file containing up to 100 coordinate pairs can be entered, preceded by a "@", and *tiny2* will generate a PSF for each position, with output filenames like test00, test01, …
- **Date of observation** *or* **PAM position of the observation and of best focus** : used to compute time- and focus-dependent aberrations and obscuration patterns

Focus & Aberrations

The NICMOS dewar began expanding when the instrument was placed in the telescope due to expansion of the nitrogen coolant. The dewar later began to contract, but it has not returned to its original state. This has placed the focus of camera 3 outside of the range of the Pupil Alignment Mechanism (PAM), which internally adjusts the NICMOS focus. Cameras 1 and 2 are also no longer confocal, but are well within the PAM range. The focus of an observation, ignoring breathing, is dependent on the PAM position at the time the image is taken, relative to the position of best focus.

Because each camera has a different focus, the PAM is moved to the best focus position for either camera 1 or 2, depending on which is the primary instrument. To bring camera 3 into focus, there were two campaigns during January and June 1998 in which the telescope's secondary mirror was moved. Otherwise, camera 3 is out of focus and typically not used. Tiny Tim includes a table of focus values for the time span of the "first life" of NICMOS and determines the proper focus for the date entered by the user (including the NICMOS 3 campaigns). The focus can also be computed from the PAM position, if the user enters "pam" instead of the date (after which it will ask for the date to determine the other aberrations).

In addition to focus, other aberrations have some effect on the NICMOS PSF, notably astigmatism, coma, and spherical aberration. These have been measured from on-orbit images using phase retrieval techniques. These vary slightly with focus, but over the focus range of science observations these changes are insignificant. They also vary with time, as the optical alignment of the NICMOS system changes (dewar expansion, FOM mirror position changes). These variations have been documented by Suchkov et al. (1999) and Krist (1999a). Tiny Tim uses the Suchkov et al. aberration/time relations for cameras 1 and 3, and the Krist relations for camera 2.

Focus, coma, and astigmatism also vary with field position. In cameras 1 and 2 focus varies linearly across the field, while in camera 3 it varies quadratically, with the best focus at the center. These variations have been measured using phase retrieval and verified by encircled energies. Camera 2 has the largest change in focus across the field – on the order of 1.5 mm equivalent PAM space defocus or about 15 μm of breathing. The changes in coma and astigmatism are fairly small in cameras 1 and 2, but are more significant in camera 3 due to its larger field. Tiny Tim asks the user for the position of the PSF on the detector and from that determines the field dependent aberrations.

Pixel Scale

Measurements of the separation of stars in a cluster have shown that the plate scale changed as the dewar expanded and contracted. However, other measurements of NICMOS 2 PSFs indicated that the effective plate scale of the PSF (defined as the radius of a given Airy ring) has remained the same. This reveals a distinction between the size of diffraction structure and the plate scale (the separation on the detector of two objects at different field positions).

Cold Mask Shifting

At the entrance to the dewar, each camera has a pupil plane mask which is intended to block the relatively warm telescope structures (spiders, region outside of the primary mirror, etc.). The masks for cameras 1 and 2 have additional tabs to cover the primary mirror support pads. In camera 2 the mask also serves as the Lyot stop of the coronagraph. The masks are cooled, hence they are called *cold masks*.

Optimally, the masks would be aligned with the telescope obscurations. But analysis of on-orbit PSFs (Krist et al. 1998) indicated that the masks are shifted, producing an asymmetric pupil function. These obscuration patterns have significant effects on the PSFs. The roughly elliptical central obscuration creates elliptical diffraction rings and distributes more light into the wings, reducing the peak pixel by 10% at 1.6 μm in camera 2. The masks in cameras 1 and 2 are shifted by about 12% of the pupil radius, and good estimates of the misalignments have been obtained. There is currently insufficient data in camera 3 to determine if there is any shifting.

The position of the cold mask (at least in camera 2) varies on both long (monthly) and short (orbital) timescales (Krist et al. 1999a). During the 1997-1998 life of NICMOS, the camera 2 mask shifted about 2% of the pupil radius, relative to an average offset of 12%. In addition, the mask can "wiggle" about this offset by about 0.5% of the pupil radius within a day. These changes cause variations in the shape of the diffraction rings and the positions of bands in the diffraction spikes of the PSF. These can result in significant residuals in PSF subtractions (see Krist et al. 1998 for examples).

Cold mask shifts for cameras 1 and 2 were implemented in version 4.4 of Tiny Tim. In version 5.0 long-term changes in the position of the camera 2 mask are included. General offsets are assumed for camera 1. Wiggle is not included, since it varies with each observation.

Filters

The available NICMOS filters are :

NICMOS 1 : F090M, F095N, F097N, F108N, F110M, F110W, F113N, F140W, F145M, F160W, F164N, F165M, F166N, F170M, F187N, F190N, POL0S, POL120S, POL240S

NICMOS 2 : F110W, F160W, F165M, F171M, F180M, F187N, F187W, F190N, F204M, F205W, F207M, F212N, F215N, F216N, F222M, F237M, POL0L, POL120L, POL240L

NICMOS 3 : F108N, F110W, F113N, F150W, F160W, F164N, F166N, F175W, F187N, F190N, F196N, F200N, F212N, F215N, F222M, F240M

Caveats

Tiny Tim produces fairly good PSFs for NICMOS, compared to the other cameras. This is mainly due to the fact that in the infrared the PSF is less sensitive to aberrations, which might not be known accurately by the modeling software. There are, however, some places where the models come up short.

Since wiggle is time dependent and apparently random, it will lead to errors between the model and real PSFs, since only a general offset for the cold mask is assumed. These errors reveal themselves as mismatches in the banding patterns in the spiders, changes in the shapes and intensities of the diffraction rings, and variations in the widths of the PSF cores. Unfortunately, observed PSFs suffer from the same problems. You can try to match the wiggle for a given observation by changing the mask x and y offset values in the parameter file produced by *tiny1* (by 0.005 or so at most, usually).

If models are being used for photometry (by shifting and normalizing to fit the observed PSFs), it may be necessary to tweak the focus to account for breathing or dewar changes. See the earlier section, *General PSF Properties*.

There are a number of electronics problems in NICMOS which produce features which may be mistaken for PSF structures. The most noticeable one is called "Stay Puft." This is seen as an increase in signal in columns containing a bright feature, like a well-exposed PSF core.

STIS

STIS (Space Telescope Imaging Spectrograph) is a combined optical imager and spectrograph, operating over the wavelength range of 120 – 1000 nm. It consists of three separate cameras – a 1024^2 CCD camera for visible light measurements and two 1024^2 MAMA detectors for far and near ultraviolet work. The CCD camera includes a coronagraph. STIS was installed during the second servicing mission, along with NICMOS.

STIS Optical Parameters

Camera	Focal Ratio	Pixel Size arcsec	Field Size arcsec
CCD	35.5	0.05079	52.0
NUV MAMA	86.9	0.02474	25.3
FUV MAMA	86.9	0.02474	25.3

Filters

Because it is primarily used as a spectrograph, STIS has a limited filter set for imaging. Note that both the CCD and MAMA cameras have clear apertures (50CCD or 25MAMA) that do not include a filter.

CCD : 50CCD, F28X50LP, F28X50OII, F28X50OIII

NUV MAMA : 25MAMA, F25CIII, F25CN182, F25CN270, F25MGII, F25QTZ, F25SRF2

FUV MAMA : 25MAMA, F25LYA, F25QTZ, F25SRF2,

Halos and Ghosts

STIS CCD PSFs have a halo that becomes more prominent at longer wavelengths. The CCD silicon substrate is mounted on glass, and at wavelengths longer than ~700 nm some photons can travel through the CCD and into the glass, where they are back-scattered into other pixels. STIS NUV PSFs have an extended halo caused by charge diffusion in the MAMA detector (it is not nearly as prominent in the FUV channel). STIS halos are not currently included in Tiny Tim due to a lack of suitable information for characterizing them.

Optical ghosts are also present in CCD images. These appear as circles positioned on a line between a star and the center of the detector. These ghosts are not included by Tiny Tim.

Charge Diffusion

Like all CCDs, some photons may be detected in pixels adjacent to those in which they fell, causing a slight blurring of the image. This is due to charge diffusion, which is described in more detail in the WFPC2 section. Tests for the ACS project on CCDs like that used in STIS have provided good measures of the charge diffusion at different wavelengths. Tiny Tim approximates the blurring caused by this phenomenon by convolving the PSF with a 3 x 3 pixel kernel appropriate for a given wavelength. This is only done for normally sampled PSFs. Subsampled PSFs should be manually convolved with the kernel AFTER the PSF is rebinned to normal sampling by the user. The appropriate kernel is provided in the FITS header for PSFs produced by *tiny2*.

Defocus

Both the CCD and MAMA imaging modes are slightly out of focus, because the instrument is optimized for spectrographic work. The amount of defocus is equal to about 4.3 μm of breathing (and is included by Tiny Tim). In addition, there is an additional 1.5 microns equivalent defocus caused by non-parfocality with the WFPC2 PC detector, which is assumed to always be at best focus.

Caveats

The CCD mode includes a Lyot stop. Because the stop is not located exactly in a plane conjugate to the exit pupil, its location relative to the OTA pupil appears to shift with field position. Tiny Tim assumes that the stop does not shift and is always at the position appropriate for the center of the field (the mask is slightly misaligned at this location, and this is accounted for by Tiny Tim).

Tiny Tim does not model the CCD coronagraphic mode.

Because of the wide wavelength range, a large number of monochromatic PSFs are generated to sample the CCD clear (50CCD) and longpass (F28X50LP) filters (107 and 57 wavelengths, respectively). Thus, it can take a while to generate a PSF.

ACS

NOTE : Because of the significant distortion in the ACS, the way Tiny Tim computes ACS PSFs is somewhat different than for the other cameras. This includes the use of a third program, tiny3, that does geometric distortion (see below).

The ACS (Advanced Camera for Surveys) is set to be installed in HST during Servicing Mission 3B in late 2001 or early 2002. It is a multiple-detector instrument with a wide field camera (WFC) consisting of two butted 2048 x 4096 CCDs (15 μm/pixel), a high resolution (HRC) 1024^2 CCD (21 μm/pix), and a STIS-like 1024^2 MAMA (25 μm/pix) for solar-blind UV imaging (SBC). The HRC includes an aberrated beam coronagraph with separate occulting spots of 0.9" and 1.5" radii.

ACS Optical Parameters

Camera	Focal Ratio	Pixel Size arcsec	Field Width arcsec
WFC	26	0.050	200
HRC	72	0.025	25
SBC	72	0.030	30

Tiny1 Queries Specific to ACS

- **Camera** : if the WFC, you must choose detector 1 or 2
- **Position of PSF on the detector** : used to compute the field-dependent aberrations; the name of a file containing up to 100 coordinate pairs can be entered, preceded by a "@", and *tiny2* will generate a PSF for each position. Note that *tiny3* will only process one PSF at a time.

Generating ACS PSFs

ACS is located away from the optical axis of HST, and so it suffers from significant distortion that is not corrected by the camera's internal optics. The projection of the camera's field-of-view onto the sky looks like a sheared rectangle – the distance along one diagonal is greater than the other. Because of this effect, it is not sufficient for Tiny Tim to generate undistorted PSFs for the ACS like it does for the other cameras. In addition, while distorting PSFs, it makes sense to include the capability to PSF-convolve and distort an arbitrary input image to simulate an ACS observation.

To handle the additional tasks, the following changes have been made to the usual Tiny Tim procedures:

tiny1 will not ask if you want to generate a subsampled PSF; that is handled by the `SUB=x` command line keyword parameter to *tiny3* (see below).

tiny2 no longer produces the final PSF; that is left to *tiny3*. The PSF produced by *tiny2* is sampled 1.3 times better than the Nyquist sampling for the shortest wavelength in the specified basspand. This allows for better interpolation results in *tiny3*. This PSF is *not* geometrically distorted. It is intended strictly for use by *tiny3* and not by the user. The *tiny2* PSF will have a "_psf.fits" extension, with filenames like `psf00_psf.fits`, `psf01_psf.fits`, etc., depending on how many multiple positions you choose to simulate.

tiny2 will create a template optional parameter file that can be used by *tiny3* to generate a PSF-convolved, geometrically-distorted scene; this file has the same rootname as the *tiny1* & *tiny2* parameter file, with a `.tt3` extension. See the section below on scene generation for proper use of this facility.

tiny3 is specifically used for ACS PSF simulation; you cannot use it for other HST cameras. It reads in the PSF produced by *tiny2* and integrates it onto distorted pixels on a distorted grid. The *tiny3* command line is :

        tiny3 parameter_file *[optional_parameter_file] [SUB=x] [POS=n]*

where **`parameter_file`** is the same filename given on the *tiny1* and *tiny2* command lines, and :

optional_parameter_file : (optional) The name of a text file containing additional parameters used for scene generation. A template file is produced by *tiny2* with the `.tt3` extension. See the section on scene generation below for more details about this file. If this parameter file is specified, it must be the second parameter on the command line.

SUB=x : (optional) Specifies the desired subsampling factor of the final PSF. The specified value must be an integer between 1 and 10. The final PSF will have pixel dimensions 1/*x* of normal in each direction. Subsampling beyond a factor of 5 is probably useless – in that case you may be better off interpolating from a 5x subsampled PSF. NOTE : The HRC and WFC CCD charge diffusion kernels are not applied to subsampled PSFs, but they are provided in the FITS header so that the user can convolve the PSF with them after rebinning the PSF to normal sampling (note that the kernel is wavelength-dependent).

POS=n : (optional) Specifies which PSF to geometrically distort if PSFs at multiple positions were generated in a single run. By default, *tiny3* will only process the first position. *n* is an integer between 1 and the number of positions.

When generating a scene rather than a PSF with *tiny3*, the SUB and POS keywords are ignored.

The filenames of the PSFs produced by *tiny3* have `.fits` extensions (e.g. `acs00.fits`, `acs01.fits`, etc.).

Coronagraphic Off-Spot PSFs

The ACS HRC has a coronagraphic mode with occulting spots and a Lyot stop. While Tiny Tim does not simulate observations of targets behind an occulting spot, it does simulate field PSFs for objects outside of the spot. The off-spot PSFs are different from the regular HRC PSFs because of the Lyot stop. All of the other characteristics of the HRC are included, however (filters, distortion, etc.).

Filters

The ACS includes a set of WFPC2 and Sloan Digital Sky Survey broadband filters, and a limited selection of narrowband filters. It does have ramp filters, which are not included in the Tiny Tim database (use a monochromatic PSF of the appropriate wavelength for these).

WFC & HRC[*] : F220W, F250W, F330W, F344N, F435W, F475W, F502N, F550M, F555W, F606W, F625W, F658N, F660N, F775W, F814W, F850LP, F892N

SBC : F115LP, F122M, F125LP, F140LP, F150LP, F165LP

[*]Some of these filters are only available with the HRC

CCD Charge Diffusion

The ACS CCDs, like all CCDs, suffer to some extent from charge diffusion. This is where some electrons photogenerated within a pixel travel into adjacent pixels before being detected. This causes a slight blurring. For a more in-depth description, see the sections on WFPC2 and STIS. Tests on ACS-like CCDs have been done to derive the blurring properties and how charge diffusion varies with wavelength.

Tiny Tim approximates charge diffusion by convolving normally sampled ACS WFC or HRC PSFs with a 3 x 3 pixel kernel appropriate for the wavelength. ***It does not apply the kernel to subsampled PSFs***, which must be manually convolved by the user after rebinning the image to normal sampling. The appropriate kernel is provided in the FITS file header produced by *tiny3*.

ACS HRC Red Halo

The ACS HRC chip has a defect that creates a halo surrounding the PSF at wavelengths > 0.6 μm. The relative proportion of flux within this halo increases with wavelength. The halo is large (many arcseconds) and can contain 10-20% of the total flux. It is caused by the CCD becoming transparent to long wavelength photons, allowing red light to pass through into the glass backing, bounce around, and bounce back into the substrate. The WFC CCDs have a metallized backing that prevents such halo formation.

An estimate of the halo is added to all HRC PSFs for wavelengths > 0.6 μm.

Field Dependent Aberrations

The ACS, notably the WFC, will have significant field dependent aberrations (focus, coma, astigmatism). Tiny Tim includes these effects as predicted by optical ray-trace models. These aberrations vary slowly with field position, so it should not be necessary to finely sample the entire field – a few arcseconds or even more between positions is probably sufficient.

Geometric Distortion

As described previously, the geometric distortion in the ACS is significant. The diffraction spikes in PSFs, for instance, will look clearly non-perpendicular. Tiny Tim includes geometric distortion effects in the processing done in *tiny3*. The distortion relations used were derived from optical ray-trace models, which should be fairly good predictions of large-scale distortions. However, there may be higher-frequency distortions caused by the detector, especially in the SBC, that have not been included due to a lack of information. The distortion coefficients are likely to change once on-orbit measurements have been done.

tiny3 properly accounts for the varying pixel area, so that the distorted PSF or scene properly represents the image on the detector before any compensation for pixel area that may occur during flat fielding.

The distortion relations used in Tiny Tim Version 6.0 are given below.

WFC :
$$V2 = 254.552 + 0.001912\,y + 0.049479\,x + 7.8968\text{e-}8\,y^2 - 2.94333\text{e-}7\,yx + 3.99467\text{e-}7\,x^2 + 3.97071\text{e-}12\,y^3 - 2.42701\text{e-}11\,y^2x - 2.69765\text{e-}12\,yx^2 - 2.18297\text{e-}11\,x^3$$

$$V3 = 254.554 - 0.0494753\,y - 0.0019787\,x + 3.98757\text{e-}7\,y^2 - 2.95496\text{e-}7\,yx + 7.73173\text{e-}8\,x^2 + 2.05943\text{e-}11\,y^3 + 2.25953\text{e-}12\,y^2x + 2.5169\text{e-}11\,yx^2 - 3.28721\text{e-}12\,x^3$$

HRC :
$$V2 = 176.691 - 0.00138315\,y - 0.0283538\,x - 9.44458\text{e-}9\,y^2 - 2.12941\text{e-}7\,yx + 6.66749\text{e-}8\,x^2 - 2.64127\text{e-}11\,y^3 + 7.16872\text{e-}12\,y^2x + 3.16406\text{e-}11\,yx^2 - 1.70420\text{e-}11\,x^3$$

$$V3 = 485.444 + 0.0252266\,y + 0.00126324\,x + 1.94016\text{e-}7\,y^2 - 5.16536\text{e-}8\,yx + 6.76348\text{e-}9\,x^2 +$$

$$5.50252e\text{-}11\ y^3 - 2.84791e\text{-}11\ y^2x - 1.91024e\text{-}11\ yx^2 - 3.46178e\text{-}12\ x^3$$

SBC :
$$V2 = 176.69 - 0.00166\ y - 0.03376\ x - 0.187e\text{-}7\ y^2 - 3.25e\text{-}7\ yx + 1.07e\text{-}7\ x^2 - 4.53e\text{-}11\ y^3 + 2.901e\text{-}11\ y^2x - 3.720e\text{-}11\ yx^2 + 8.315e\text{-}11\ x^3$$

$$V3 = 485.442 + 0.03015\ y + 0.00139\ x + 31.99e\text{-}8\ y^2 - 7.54e\text{-}8\ yx + 1.83e\text{-}8\ x^2 + 6.284e\text{-}11\ y^3 - 1.444e\text{-}11\ y^2x - 4.0e\text{-}11\ yx^2 - 9.757e\text{-}11\ x^3$$

In the above equations, x and y are the distances in detector pixels from the center of the camera (x,y = 0,0 at the center of the camera). In the WFC, the camera center is between the two chips (note that Tiny Tim assumes that both detectors are perfectly adjacent to one another; in reality, there is about a 50 pixel gap between them and they are slightly rotated relative to each other). The resulting V2 and V3 values from these equations are in arcseconds on the sky in the HST coordinate system. See the diagram below for mappings from detector X,Y to V2,V3.

The coordinates on the sky (in the V2,V3 system) of the corners of the distorted PSF's center pixel are given in the FITS header and can be used to determine the pixel scale.
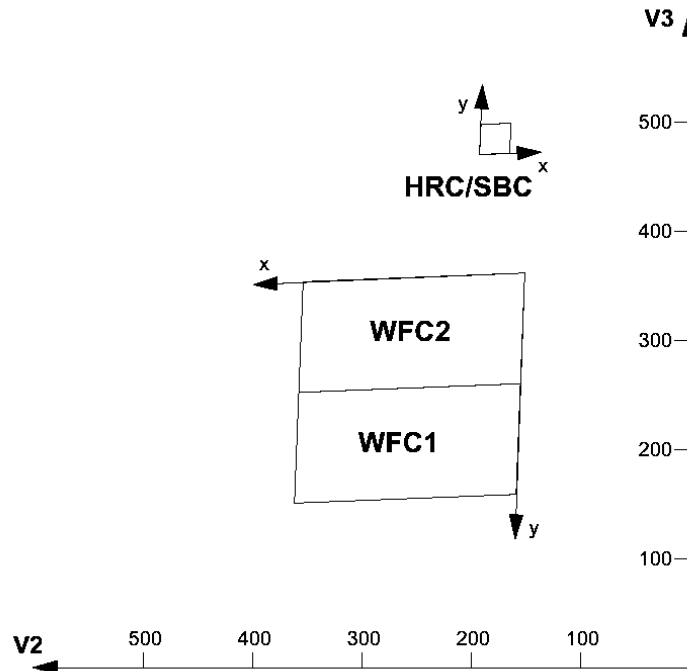
(Above) The mapping of ACS camera field-of-views on the HST focal plane (looking at the sky). The detector X,Y axis directions are indicated. V2 and V3 are shown in arcseconds relative to the OTA center. Note that each WFC has (X,Y) = (0,0) at the first pixel of the detector. From the *ACS Instrument Handbook Version 1.0*.

Scene Generation

Since *tiny3* already makes use of the PSF and can do accurate geometric distortion, it made sense to incorporate the capability to take an arbitrary input scene, convolve it with the PSF, and then distort it. Tiny Tim uses the following procedure when processing scenes :

> 1) The scene is read from the FITS file specified by the user. The PSF produced by *tiny2* is also read in.
>
> 2) If the input scene has a different pixel scale than the input PSF, it is resampled onto a new grid using damped sinc interpolation to match the PSF scale. The new grid has a maximum size of 4K by 4K pixels.
>
> 3) The input scene is convolved with the PSF.
>
> 4) The PSF-convolved scene is mapped onto a distorted grid using damped sinc interpolation. Each pixel in the distorted grid is divided into many subpixels to account for undersampling.
>
> 5) The PSF-convolved, geometrically-distorted scene is written out to a FITS file.

To use the scene processing capability of *tiny3*, you must provide an optional parameter file. A template version of this file is produced by *tiny2* with the `.tt3` extension. The format of this file looks something like this :

```
INPUT_FILENAME=no_file
PIXEL_SCALE=0.02133
X_CENTER=512
Y_CENTER=512
OUTPUT_FILENAME=no_file
```

The five keywords shown <u>must</u> be present in any optional parameter file. The default "no_file" values indicate that no scene generation is done. The meaning of these keywords are :

`INPUT_FILENAME` :
> The full name (including extension) of a FITS image file that contains the undistorted input scene. The FITS file MUST be in the standard 4-byte IEEE single-precision floating-point format (scaled integer FITS files are not supported). As described later in this section, this entry may be used to specify a blank input field rather than an input image.
> Example use : `INPUT_FILENAME=galaxy_field.fits`

`PIXEL_SCALE` :
> The scale, in arcseconds per pixel, of the image specified by `INPUT_FILENAME`. By default, this is set to the scale of the PSF produced by *tiny2*. The user must set this entry to the appropriate value for the input image. The image will be resampled via damped sinc interpolation onto a grid with the same sampling as the *tiny2* PSF, so the input scene should not be so coarsely sampled that significant interpolation artifacts may occur. The input image should also not be so large that it exceeds the limits of the largest available grid (4K x 4K) when resampled to the scale of the input PSF, otherwise the edges of the field will be truncated.

X_CENTER, Y_CENTER :

> The corresponding location on the detector, in pixels, of the scene image center. Note that the first pixel on the detector is (0,0). If you generated PSFs at multiple positions, then *tiny3* will use the PSF closest to this position.

OUTPUT_FILENAME :

> This is the name of the FITS file that will contain the PSF-convolved, geometrically-distorted version of the input scene. You should provide the extension.

tiny3 can also add PSFs of specified fluxes to the scene at specified locations. This facility should be used rather than attempting to add PSFs by having delta functions in the input scene (which would probably be distorted by the resampling step). To add a PSF, include the following line in the optional parameter file:

> **PSF=** *x_position y_position counts*

where *x_position* and *y_position* are offsets from the scene center in arcseconds, and *counts* is the total counts in the PSF. Multiple PSF lines can be included in a single optional parameter file.

You can also create a scene with just PSFs by using the blank field option for the INPUT_FILENAME and using multiple PSF entries.   To create a blank field, enter blank_*xsize_ysize* as the INPUT_FILENAME, where xsize and ysize are the dimensions of the field in arcseconds. For example, to create a 4.5 by 4.5 arcsecond field and then add a few PSFs of various intensities, you could use something like the following :

```
INPUT_FILENAME=blank_4.5_4.5
PIXEL_SCALE=0.0
X_CENTER=512
Y_CENTER=512
OUTPUT_FILENAME=psfs.fits
PSF=  0.0  0.0  1500.0
PSF= -1.5 -1.5  1000.0
PSF=  1.5 -1.5  2000.0
PSF=  1.5  1.5  3000.0
PSF= -1.5  1.5  4000.0
```

The value of the PIXEL_SCALE entry is ignored (but must still be present) when the blank field option is used.

You should not try to create a single 4K by 2K WFC image by using a single PSF and a large input scene. The scene processing facility is meant to be used on small regions that the user will manually mosaic into a full image. Thus, you should attempt to include overlapping regions in your input scene.

FOC

The Faint Object Camera is the only remaining first-generation science instrument left in HST. It is optimized for UV studies with a wavelength range of 100 – 500 nm. It is now corrected for spherical aberration in the telescope by COSTAR (Corrective Optics Space Telescope Axial Replacement). The FOC consists of two separate detectors, commonly referred to as the f/96 and f/48 channels. Due to previous electronics problems, the f/48 channel is rarely used.

FOC (Pre-COSTAR) Optical Parameters[1]

Camera	Focal Ratio	Pixel Size arcsec	Field Size arcsec
f/96	96	0.022	11.4
f/48	48	0.045	22.9

[1] For 512 x 512 pixel mode (unzoomed)

The addition of COSTAR altered the optical characteristics of the FOC:

FOC (with COSTAR) Optical Parameters[1]

Camera	Focal Ratio	Pixel Size arcsec	Field Size arcsec
f/96	151	0.014	7.3
f/48	72.5	0.030	15.2

[1] For 512 x 512 pixel mode (unzoomed)

Tiny Tim Queries Specific to FOC

- **Date of observation (pre-COSTAR) :** used to compute the focus and aberration values, which are time dependent

Image Sizes

The FOC has selectable image sizes of 512 x 512 pixels or 512 x 1024. In the latter mode, the dynamic range is only 8-bits (256 levels). One dimension of a pixel can be stretched by a factor of two (zoomed mode). Tiny Tim only models the 512 x 512 modes, using the optical parameters listed in the tables above.

Distortion and Field Dependent Aberrations

The images produced by the FOC are highly distorted due to the nature of the photon counting detectors. Most of the distortion is corrected in the calibrated data, but near the field edges it is still fairly high. One should only assume that the central 256 x 256 region is well corrected. Tiny Tim assumes that there is no distortion. There is also some field dependence of the aberrations, but they are small and not well characterized and thus are not included in Tiny Tim FOC models.

Filters

The available FOC filters in Tiny Tim are:

f/48 : F130LP, F140W, F150W, F175W, F180LP, F195W, F220W, F275W, F305LP

f/96 : F120M, F130M, F140M, F152M, F165W, F170M, F175M, F190M, F195M, F210M, F220W, F231M, F253M, F275W, F278M, F307M, F320W[*], F342W, F346M, F370LP, F372M, F410M, F430W, F437M, F470M, F480LP, F486N, F501N, F502N, F550M, F600M, F630M

[*] The F320W filter appears to significantly degrade the PSF for unknown reasons. This degradation is *not* modeled by Tiny Tim.

Caveats

Since the FOC typically operates at short wavelengths, it is very sensitive to the aberrations in the telescope, especially the mirror zonal errors, which cause the PSF to change dramatically with wavelength (as shown above). Uncertainties in the zonal errors and other aberrations will result in mismatches between the model and observed PSFs, especially in the wings of the aberrated images.

As with WF/PC-1, the aberrated FOC PSFs were very sensitive to focus and other aberration changes.

WF/PC-1

The first Wide Field/Planetary Camera (WF/PC-1) was an eight-channel instrument – four wide-field cameras (WFC 1-4) or four planetary cameras (PC 5-8), with either set selected using a rotating pyramid mirror.

WF/PC-1 Optical Parameters

Camera	Focal Ratio	Pixel Size arcsec	Field Size arcsec
WFC 1-4	12.9	0.1	76
PC 5-8	30.0	0.043	33

Tiny1 Queries Specific to WF/PC-1

- **Camera :** either 1-4 for the WFCs or 5-8 for the PCs
- **Position on the detector :** used to compute the obscuration pattern; the name of a file containing up to 100 coordinate pairs can be entered, preceded by a "@", and *tiny2* will generate a PSF for each position, with output filenames like test00, test01, …
- **Date of observation :** used to compute focus and aberrations, which are time dependent

Field Dependence and Aberrations

Like WFPC2, the obscurations in the WF/PC-1 optics appeared to shift relative to the telescope's depending on field position. The effect of this shifting was more pronounced in WF/PC-1 because of the spherical aberration, which caused "shadows" of the camera obscurations and the telescope mirror support pads. The increased beam size due to the aberration also caused vignetting at the edges of the field in the WFCs.

These variations made deconvolution difficult, since the positions of the shadows in the wings could change significantly with field offsets of only tens of pixels. The resulting mismatches would lead to residuals in the wings of the PSFs. Since Tiny Tim models can be generated for any position, an optimum PSF can be used for wherever the object is on the detector.

WF/PC-1 was known to have field dependent aberrations (focus, coma, and astigmatism), but they were not well characterized. The aberrations for the center of the field in each camera are used for all positions in Tiny Tim. WF/PC-1 also suffered from geometric distortion, which is *not* modeled.

The WF/PC-1 detectors must have had charge diffusion blurring like that in WFPC2, but its effect was not noticeable due to the blurring caused by spherical aberration. Charge diffusion effects are *not* included in Tiny Tim WF/PC-1 models.

Filters

The following WF/PC-1 filters are available in Tiny Tim:

F122M	F157W	F194W	F230W	F284W
F336W	F368M	F375N	F413M	F437N
F439W	F469N	F487N	F492M	F502N
F517N	F547M	F555W	F569W	F588N
F606W	F622W	F631N	F648M	F656N
F658N	F664N	F673N	F675W	F702W
F718M	F725LP	F785LP	F791W	F814W
F850LP	F875M	F889N	F1042M	F1083N

Caveats

WF/PC-1 PSFs are highly dependent of field position due to the shifting obscuration patterns. Also, because the spherical aberration creates shadows of the spiders, central obscuration, and mirror support pads, the PSFs are very sensitive to focus changes and optical distortions. The focus for a given observation may need to be altered by adjusting the focus coefficient in the parameter file produced by *tiny1*, as discussed previously.

Appendix A

References

Additional information on HST PSFs can be obtained from the STScI web site (www.stsci.edu), including instrument science reports (ISRs), instrument handbooks, and advisories.

Hasan, H. & Burrows, C.J. (1995) "Telescope Image Modeling (TIM)." P.A.S.P. 107, 289.

Krist, J.E. (1999a) "Time dependence of the NICMOS 2 aberrations and obscuration pattern." NICMOS ISR 99-011

Krist, J.E. & Burrows, C.J. (1995) "Phase retrieval analysis of pre- and post-repair Hubble Space Telescope images." Applied Optics, 34, 4951.

Krist, J.E. & Burrows, C.J. (1994) "Large angle scattering in WFPC2 and horizontal smearing correction." WFPC2 ISR 94-01.

Krist, J.E. & Hook, R. (1997) "NICMOS PSF variations and Tiny Tim simulations." Proceedings of the 1997 HST Calibration Workshop, p. 192.

Krist, J.E. et al. (1998) "Characterization and subtraction of well-exposed HST/NICMOS Camera 2 point spread functions for a survey of very-low-mass companions to nearby stars." P.A.S.P. 110, 1046.

Lauer, T. (1999) "The photometry of undersampled point spread functions." P.A.S.P., 111, 1434

Storrs, A. et al. (1999) "Camera 3 Intrapixel Sensitivity." NICMOS ISR 99-005.

Suchkov, A. & Galas, G. (1999) "NICMOS optical aberrations : coma and astigmatism." NICMOS ISR 99-003.

Suchkov, A. & Krist, J. (1998) "Tiny Tim modeling of NICMOS PSF subtraction: effects of focus breathing, cold mask wiggling, and color variations." NICMOS ISR 98-018

Appendix B

How PSFs are Simulated

The following is a brief description of how Tiny Tim simulates PSFs. For a more thorough discussion of HST optics and diffraction optics, check out *Astronomical Optics* by Daniel Schroeder.

A PSF is the diffraction pattern caused by light from a point source passing through an optical system, with the various optical components and obscurations adding structure to it. In the case of HST, the obscurations include the entrance aperture, the secondary mirror and spider vanes, the support pads on the primary mirror, the WFPC repeater cameras' secondaries and spiders, and any vignetting which occurs from other sources (like the WFC entrance aperture). Optical aberrations manifest themselves as errors in the optical path length. For HST these include spherical aberration, defocus, coma, astigmatism, clover, and mirror zonal aberrations, along with others.

These properties can be simulated on a computer by creating a two dimensional array which contains the aperture function (basically a mask which shows where the beam is not being obscured) and another array containing the optical path differences (OPD) function. The aperture function depends on which camera is being used, and if it is WFPC, the shifts caused by the repeater camera obscurations must also be taken into account. The OPD can be calculated as the sum of Zernike polynomials, which describe specific aberrations, and additional terms, such as local errors in the mirrors' surfaces. A list of Zernike polynomials is included in the *HST OTA Handbook*.

These two functions are generated for each point on a grid of Nx/2 by Ny/2, where Nx and Ny are the dimensions of the arrays. This provides for critical sampling of the functions. Since Tiny Tim uses square arrays, the functions are defined only for the central N/2 by N/2 region. The array size depends on how large a PSF is needed.

The OPD and aperture function, A, form the complex pupil function:

$$P = Ae^{(2\pi i OPD/\lambda)}$$

and the Fourier transform of the pupil function is the amplitude spread function (ASF). Finally, the modulus squared of the ASF is the PSF:

$$PSF = \left| FFT(P) \right|^2$$

This is the Nyquist-sampled PSF for a specific wavelength (in which the OPD is defined), with the spacing between pixels being $F\lambda/2$, where F is the focal ratio of the instrument (such as 12.9 for the Wide Field Camera). Thus, the sizes of the arrays used to generate the critical PSF are dependent on the required size of the integrated PSF and the wavelength. Shorter wavelengths require larger grid sizes, and thus more time to compute.

The Nyquist-sampled PSF is then Fourier transformed to produce the optical transfer function (OTF). If jitter is specified, the OTF is multiplied by a function that models Gaussian jitter.

Integration onto detector pixels is performed by multiplying the OTF by the analytical expression for the integrated detector OTF. The result is then Fourier transformed back and the integrated values sampled using damped-sinc interpolation.

Because the PSF changes with wavelength, a number of PSFs have to be generated by the above method at wavelengths which sample the filter being used, to account for the filter's bandpass. The monochromatic models are combined via a weighted addition. The weights are dependent on the optical system throughput, including the filter, and the object spectrum.

Appendix C

Object Spectra and Filter Curves

Because the size and shape of the HST PSF is dictated by diffraction rather than seeing effects, you must pay attention to how the PSF varies with wavelength. As wavelength increases, the diffraction structure (rings, spikes) expand proportionally. In the same broad band filter, the PSF of a red star may be noticeably larger than that of a more blue one. If you subtract a red object PSF from a blue object's, you can end up with significant residuals (this only applies to wide band filters – the change over the bandpass of a narrow filter is far too small to be significant).

Since object color can make a difference, you should try to match the object spectrum when you generate Tiny Tim PSFs in wide band filters. Tiny Tim offers a selection of 17 different object colors (O through M stars), using selected spectra from the Bruzual-Peterson-Gunn-Stryker spectrophotometric atlas. One of these is likely to match your object's color in a given bandpass. You can also specify a blackbody spectrum, if that is appropriate.

With whatever spectrum option you choose (including your own), you can also apply standard interstellar extinction by using the **ebmv** or **Av** keyword when running *tiny1* (see *Command Line Parameters* on p. 8).

Specifying Your Own Spectrum

If your object is so different that it cannot be reasonably well represented by one of the default spectra, then you also have the option of providing Tiny Tim a spectrum to use instead. You do not need to worry about this if you are using anything other than a wide band (or long pass) filter, or if your spectrum is reasonably matched by one of the default Tiny Tim spectra over the filter passband.

You must provide an ASCII text file which contains the wavelength (in Angstroms) and corresponding flux, one pair per line, up to 15000 pairs. The first line of the file MUST be one of the following, which tells Tiny Tim what units the fluxes are in : FLAM, FNU, PHOTLAM, or JY, where these are defined as :

FLAM	ergs Angstrom^{-1}
FNU	ergs Hz^{-1}
PHOTLAM	photons Angstrom^{-1}
JY	Janskys (W Hz^{-1})

The normalization of the fluxes is not important, only the relative differences between wavelengths. Your spectrum <u>must not</u> include camera system response effects (these are handled when you specify the filter). Tiny Tim will interpolate your spectrum to determine the flux for each component monochromatic PSF used to generate a polychromatic PSF.

Note that specifying the wavelength in Angstroms is a departure from the rest of Tiny Tim, where wavelength is usually given in nanometers. The wavelength range of your spectrum should cover the range of the filter you will be using, and then some (otherwise, the spectrum will be extrapolated).

The beginning of a spectrum file might look something like this :

```
PHOTLAM
100.0  1.0e-3
120.0  1.3e-3
130.0  1.5e-3
…
```

You can generate an appropriate spectrum using SYNPHOT in STSDAS. For example, suppose you want to provide a 5000 K black body spectrum (even though you can do it in Tiny Tim). You can use the SYNPHOT `calcspec` routine like so to write the spectrum to the file `my_spectrum.tab` :

`calcspec bb(5000) my_spectrum form=photlam`

Note that the output here is in photlam (photons/cm^2/sec/Angstrom). `calcspec` uses Angstroms for the wavelength. If the default wavelength range used by `calcspec` doesn't cover the range of your filter, you may need to provide a wavelength file using the `wavetab` keyword (see the SYNPHOT manual for more information).

To convert the STSDAS *.tab table file to an ASCII file, use the STSDAS `tdump` command :

`tdump my_spectrum.tab > my_spectrum.txt`

Then, you have to edit the beginning of the text file to get rid of the header lines and then put the form of the fluxes (in this case PHOTLAM) in the first line. Tiny Tim will only read the first two columns of the spectrum and ignore any remaining ones.

The filename of your spectrum table is given to Tiny Tim using the **spectrum** keyword when running **tiny1**. For example :

`tiny1 test.in spectrum=my_spectrum.txt`

You can provide separate spectrum and filter tables, if you wish.

Specifying Your Own Filter Table

If for some reason Tiny Tim's default filter throughput tables are not sufficient, you can provide your own.

To do so, you must create a text file containing the wavelength (in nanometers) and corresponding transmission, one pair per line, in order of ascending wavelength. A maximum of 200 wavelengths can be entered. The wavelengths should be spaced by about $0.018\lambda_{nm}$. The object spectrum, filter throughputs, and detector response all go into computing the transmission.

The filename of your filter table is given to Tiny Tim using the **filter** keyword when running **tiny1**. For example :

`tiny1 test.in filter=my_filter.txt`

You can provide separate spectrum and filter tables, if you wish.